Applied Mathematics and Computation 333 (2018) 231–245

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Third derivative modification of k-step block Falkner methods for the numerical solution of second order initial-value problems

Higinio Ramos[a,b,*], M.A. Rufai[c]

[a] Scientific Computing Group, Universidad de Salamanca, Plaza de la Merced, Salamanca 37008, Spain
[b] Escuela Politécnica Superior de Zamora, Campus Viriato, Zamora 49022, Spain
[c] Department of Mathematical Sciences, The Federal University of Technology, Akure P.M.B., Akure 704, Ondo State, Nigeria

ARTICLE INFO

Keywords:
Falkner-type methods
Second-order differential equations
Block methods
Stability analysis

ABSTRACT

This paper is devoted to the development and analysis of a modified family of Falkner-type methods for solving differential systems of second-order initial-value problems. The approaches of collocation and interpolation are adopted to derive the new methods. These modified methods are implemented in block form to obtain the numerical solutions to the considered problems. The study of the properties of the proposed block Falkner-type methods reveals that they are consistent and zero-stable, and thus, convergent. From the stability analysis, it could be seen that the proposed Falkner methods have non-empty stability regions for $k = 2, 3, 4$. Some numerical test are presented to illustrate the efficiency of the proposed family.

1. Introduction

This paper aims at obtaining numerical solutions for second-order initial-value problems (IVP) of the general form

$$y''(x) = f(x, y(x), y'(x)), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0. \tag{1}$$

In most cases, the mathematical formulation of the physical phenomena in Engineering, Mathematical Biology, Epidemiology, Physics and Economics among other disciplines are carried out through problems of the form in (1). Usually, these problems do not have known exact solutions, hence there is a need for approximate solutions. Conventionally, in the past, any problem of the form in (1) was often solved by reducing it to an equivalent system of first-order ODEs and then applying one of the different approaches available for solving systems of first order IVPs. Reduction methods have been discussed in detail by many notable researchers like Lambert [1,2], Onumanyi et al. [3], and Jeltsch and Kratz [4], to mention but a few. It was reported in Awoyemi [5] that writing computer code for solving the equivalent system required large human effort and was time-consuming, compared with the appropriate formulation for solving the problem directly.

In order to get more accurate numerical solutions with less effort to solve the second-order differential equation, many scholars including Vigo-Aguiar and Ramos [6], Mazzia and Nagy [7], Mazzia et al. [8], Sommeijer [9], Brugnano and Trigiante [10], Butcher and Hojjati [11], Franco [12], Mahmoud and Osman [13], among others, developed different numerical

[*] Corresponding author.
E-mail addresses: higra@usal.es (H. Ramos), mrufai@aims.ac.tz (M.A. Rufai).

https://doi.org/10.1016/j.amc.2018.03.098

methods to give the approximate solutions to (1) and higher-order ODEs without reducing it to a system of first-order ODEs. Some of the numerous numerical approaches presented by the aforementioned researchers included higher derivative multistep methods, Runge–Kutta methods, spline-collocation methods, and Runge–Kutta–Nyström methods. Although their implementations yield good accuracy, the procedures are not easy to implement as the subroutines involved are very much complicated, since they require special techniques to supply the starting values and for varying the step size.

The need to improve on the aforementioned methods became important to researchers in this area. Different scholars such as Jator [14], Adeniran and Ogunware [15], Jator and Li [16], Ramos and Lorenzo [17], Ramos and Patricio [18], Yap et al. [19], Ramos et al. [20], Rufai and Areo [21], have developed block methods for solving higher-order ODEs directly. In block methods, the approximations are simultaneously obtained at a number of consecutive grid points in the interval of integration. These methods are less costly in terms of number of function evaluations compared to the reduction method and linear multistep methods. A merit of the block methods over traditional predictor-corrector ones is that they give better approximate solutions when solving many problems of the form in (1) and higher-order ODEs directly. Recently, Li and Wu [22], Ramos et al. [23] have adopted Falkner block methods to give a direct solution to (1). In this manuscript, we develop a family of new Falkner-type block methods using the third derivative to get directly numerical solutions for problems of the form in (1). The third derivative may be obtained easily using a computer algebra system, and we will get that this contribution results in an increase in the order of the methods and much better stability regions (compared with the methods with the same number of equations in [23] which do not consider the third derivative).

The paper is organized as follows. In Section 2, we present the procedure to obtain the k-step modified block Falkner methods for solving the problem in (1). The analysis of the methods is discussed in Section 3. In Section 4 we present some particular cases of the family of block methods developed previously and implementation details are well explained in Section 5. Some numerical examples are given in Section 6 to show the efficiency of the proposed methods. Finally, some conclusions are discussed in Section 7.

2. Derivation of the modified block Falkner methods

As in many numerical methods, for solving the problem in (1) we assume that we can approximate its solution $y(x)$ by a polynomial $p(x)$. We consider $k+1$ grid points given by $x_n, x_{n+1}, \dots, x_{n+k-1}, x_{n+k}$ with a fixed stepsize $h = x_{j+1} - x_j$. Let us obtain the block formulation for a k-step modified Falkner-type method considering a third derivative, and then we will discuss the characteristics of the methods, and will present the formulas for some particular values of k. For this, let us assume that the solution $y(x)$ of (1) is approximated by the polynomial $p(x)$ in the form

$$y(x) \simeq p(x) = \sum_{j=0}^{k+3} a_j x^j. \tag{2}$$

where the a_j, with $j = 0, \dots, k+3$, are $k+4$ unknowns that will be determined.

Therefore, we have that

$$y'(x) \simeq p'(x) = \sum_{j=1}^{k+3} j a_j x^{j-1}, \tag{3}$$

$$y''(x) \simeq p''(x) = \sum_{j=2}^{k+3} j(j-1) a_j x^{j-2}. \tag{4}$$

We impose that the approximating polynomial in (2) and its first derivative in (3) coincide with the approximate solution and its approximate derivative at the point x_{n+k-1}, and that the second derivative in (4) coincides with the approximate second derivative of the solution at the points x_{n+j}, $j = 0, 1, \dots, k$. They all make a total of $k+3$ equations. We need one more equation, and we consider the approximation of the third derivative of the solution at the point x_{n+k}, which is written by $g_{n+k} \simeq y'''(x_{n+k})$. In this way we obtain a system of $k+4$ algebraic equations with $k+4$ unknowns (the a_j, $j = 0, 1, \dots, k+3$) given by

$$\begin{cases} y_{n+k-1} = p(x_{n+k-1}) \\ y'_{n+k-1} = p'(x_{n+k-1}) \\ f_{n+j} = p''(x_{n+j}), \qquad j = 0, 1, \dots, k \\ g_{n+k} = p'''(x_{n+k}) \end{cases},$$

where as usual, the values $y_{n+j}, y'_{n+j}, f_{n+j}, g_{n+j}$ are notations for the approximate values of $y(x_{n+j}), y'(x_{n+j}), f(x_{n+j}, y(x_{n+j}), y'(x_{n+j}))$ and $f'(x_{n+j}, y(x_{n+j}), y'(x_{n+j}))$ respectively.

This system may be written in matrix form as

$$MA = B$$

where

$$M = \begin{pmatrix} 1 & x_{n+k-1} & x^2_{n+k-1} & x^3_{n+k-1} & \cdots & x^{k+3}_{n+k-1} \\ 0 & 1 & 2x_{n+k-1} & 3x^2_{n+k-1} & \cdots & (k+3)x^{k+2}_{n+k-1} \\ 0 & 0 & 2 & 6x_n & \cdots & (k+3)(k+2)x_n^{k+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 2 & 6x_{n+k} & \cdots & (k+3)(k+2)x^{k+1}_{n+k} \\ 0 & 0 & 0 & 6 & \cdots & (k+3)(k+2)(k+1)x^k_{n+k} \end{pmatrix},$$

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{k+2} & a_{k+3} \end{pmatrix}^T,$$

$$B = \begin{pmatrix} y_{n+k-1} & y'_{n+k-1} & f_n & \cdots & f_{n+k} & g_{n+k} \end{pmatrix}^T.$$

The solution of the above system gives us the required coefficients of the polynomial $p(x)$ in terms of $y_{n+k-1}, y'_{n+k-1}, f_n, f_{n+1}, \ldots, f_{n+k}$ and g_{n+k}.

Now, in order to form the block method we evaluate $p(x)$ and $p'(x)$ at the point x_{n+k}, obtaining the two main formulas of the block method. The other additional formulas of the block method are obtained through the evaluation of $p(x)$ and $p'(x)$ at the points $x_{n+j}, j = 0, 1, \ldots, k-2$. All this makes a total of $2k$ equations with $2k$ unknowns (these are the approximate values of the solution and the derivative at the grid points). The following $2k$ equations become those equations and constitute what we have called the modified k-step block Falkner method

$$\begin{cases} p(x_{n+k}) = y_{n+k}, \\ p'(x_{n+k}) = y'_{n+k}, \\ p(x_{n+j}) = y_{n+j}, \qquad j = 0, \ldots, k-2, \\ p'(x_{n+j}) = y'_{n+j}, \qquad j = 0, \ldots, k-2. \end{cases} \tag{5}$$

After solving the above system of $2k$ equations in the $2k$ unknowns y_{n+i}, y'_{n+i} for $i = 1, \ldots, k$ we obtain the approximate values for the solution and its derivatives at the k grid points $x_{n+j}, j = 1, 2, \ldots, k$. To solve the system, depending on whether it is linear or not, one can use some of the numerical algorithms that exist for this purpose. In the case of non-linear systems, the Newton-type methods are widely used.

Note that for the special second order equation $y'' = f(x, y)$, if we are not particularly interested in obtaining the approximation of the derivatives, the number of equations in the above system may be decreased. Indeed, the above system, that is, the modified Falkner method for $y'' = f(x, y)$, may be rewritten as

$$\begin{cases} p(x_{n+k}) = y_{n+k} \\ p'(x_{n+k}) = y'_{n+k} \\ p(x_{n+j}) = y_{n+j}, \qquad j = 0, \ldots, k-2 \\ p'(x_n) = y'_n \end{cases}, \tag{6}$$

where there are only $k+2$ equations and $k+2$ unknowns $(y_{n+1}, \ldots, y_{n+k}$, and $y'_{n+k-1}, y'_{n+k})$. In order to apply the block method at the next block the only starting values we need are the values of the solution and the derivative at the final point of the previous block. On the other hand, the unique derivative that appears in the formulas that approximate the solutions is y'_{n+k-1}. Thus, just considering two formulas including these unknown derivatives, given by $p'(x_{n+k}) = y'_{n+k}$ and $p'(x_n) = y'_n$, are enough to proceed. For differential equations where the function f does not contain the first derivative, this approach represents a considerable saving in computational time.

3. Analysis of the methods

Here, we consider the analysis of the basic properties of the proposed methods, which includes: zero stability, order, local truncation error, region of absolute stability, consistency, and convergence of the methods.

3.1. Local truncation error and order

The above block method in (5) may be written in the form

$$A Y_n = h B Y'_n + h^2 C F_n + h^3 D G_n \tag{7}$$

Table 1
Order and error constants for the formulae in the two-step method.

Schemes	Order	Error constants
y_{n+2}	4	$\frac{1}{360}$
y_n	4	$-\frac{1}{144}$
y'_{n+2}	4	$\frac{23}{1440}$
y'_n	4	$\frac{7}{1440}$

where A, B, C, D are matrices of coefficients with dimensions $2k \times (k+1)$, and

$$\begin{aligned} Y_n &= (y_n, y_{n+1}, \ldots, y_{n+k},)^T, \\ Y'_n &= \left(y'_n, y'_{n+1}, \ldots, y'_{n+k},\right)^T, \\ F_n &= (f_n, f_{n+1}, \ldots, f_{n+k},)^T, \\ G_n &= (g_n, g_{n+1}, \ldots, g_{n+k},)^T. \end{aligned}$$

For example, for $k = 2$ the corresponding matrices are given by

$$A = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} \frac{1}{80} & -\frac{3}{10} & -\frac{17}{80} \\ \frac{1}{48} & -\frac{5}{12} & -\frac{29}{48} \\ -\frac{23}{240} & -\frac{8}{15} & \frac{31}{240} \\ \frac{17}{48} & \frac{11}{12} & -\frac{13}{48} \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 & \frac{7}{120} \\ 0 & 0 & \frac{1}{8} \\ 0 & 0 & -\frac{7}{120} \\ 0 & 0 & \frac{1}{8} \end{pmatrix}.$$

Following the guidelines proposed by Ramos et al. in [23], assuming that $y(x)$ is a sufficiently differentiable function, we define the linear difference operator $\mathbb{L}$ associated to the block method in (5) as follows:

$$\mathbb{L}[y(x); h] = \sum_{j=0}^{k} \left[\bar{\alpha}_j y(x_n + jh) - h\bar{\beta}_j y'(x_n + jh) - h^2 \bar{\gamma}_j y''(x_n + jh) - h^3 \bar{\delta}_j y'''(x_n + jh) \right] \tag{8}$$

where $\bar{\alpha}_j, \bar{\beta}_j, \bar{\gamma}_j$ and $\bar{\delta}_j$ are respectively the vector columns of matrices A, B, C and D.

On expanding Eq. (8) using Taylor series about x_n we get the following equation.

$$\mathbb{L}[y(x); h] = \bar{C}_0 y(x_n) + \bar{C}_1 h y'(x_n) + \bar{C}_2 h^2 y''(x_n) + \cdots + \bar{C}_q h^q y^q(x_n) + \cdots , \tag{9}$$

where

$$\bar{C}_q = \frac{1}{q!} \left[\sum_{j=1}^{k} j^q \bar{\alpha}_j - q \sum_{j=1}^{k} j^{q-1} \bar{\beta}_j - q(q-1) \sum_{j=1}^{k} j^{q-2} \bar{\gamma}_j - q(q-1)(q-2) \sum_{j=1}^{k} j^{q-3} \bar{\delta}_j \right], \tag{10}$$

where $q = 0, 1, 2, 3, \ldots$.

Definition 3.1. The linear difference operator and the associated modified block method are said to be of order p if $\bar{C}_0 = \bar{C}_1 = \cdots = \bar{C}_{p+1} = 0, \bar{C}_{p+2} \neq 0$. The $\bar{C}_i$ are column vectors of scalars of size $2k$, and $\bar{C}_{p+2}$ is the vector of error constants.

The order and error constant of the proposed methods for $k = 2, 3, 4$ are given respectively in Tables 1–3. These values correspond to the formulas of the methods as they appear in Section 4.

Table 2
Order and error constants for the formulae in the three-step method.

Schemes	Order	Error constants
y_{n+3}	5	$-\frac{1}{175}$
y_{n+2}	5	$-\frac{41}{16800}$
y_{n}	5	$\frac{97}{16800}$
y'_{n+3}	5	$-\frac{1}{450}$
y'_{n+2}	5	$-\frac{11}{2400}$
y'_{n}	5	$-\frac{97}{7200}$

Table 3
Order and error constants for the formulae in the four-step method.

Schemes	Order	Error constants
y_{n+4}	6	$\frac{27}{4480}$
y_{n+3}	6	$\frac{1}{252}$
y_{n+2}	6	$\frac{209}{120960}$
y_{n}	6	$-\frac{3}{640}$
y'_{n+4}	6	$\frac{3}{1120}$
y'_{n+3}	6	$\frac{1}{756}$
y'_{n+2}	6	$\frac{97}{30240}$
y'_{n}	6	$\frac{337}{30240}$

3.1.1. Zero-stability

The concept of zero-stability is concerned with the behavior of the difference system (7) in the limit as h tends to zero. Thus, as $h \to 0$, the method in (7) results in the following system of equations

$$\begin{cases} y_{n+1} = y_{n+k-1} \\ y_{n+2} = y_{n+k-1} \\ \quad \dots \\ y_{n+k-2} = y_{n+k-1} \\ y_{n+k-1} = y_n \\ y_{n+k} = y_{n+k-1} \end{cases}$$

which may be rewritten in matrix notation as

$$Id\,\bar{Y}_\mu - \bar{A}\,\bar{Y}_{\mu-1} = 0$$

where

$$\bar{Y}_\mu = (y_{n+1}, y_{n+2}, \dots, y_{n+k})^T, \qquad \bar{Y}_{\mu-1} = (y_n, y_{n+1}, y_{n+k-1})^T$$

with Id the identity matrix of order k and $\bar{A}$ the matrix of order k given by

$$\bar{A} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}.$$

The block method is zero-stable provided the roots R_j of the first characteristic polynomial $\rho(R)$ given by $\rho(R)=\det[Id\,R-\bar{A}]$ satisfy $|R_j|\leq 1$, and for those roots with $|R_j|=1$ the multiplicity does not exceed 2 (see [24]). Since $\rho(R)=(R-1)R^{k-1}$, the block method in (7) is zero-stable.

3.2. Consistency and convergence of the proposed methods

According to Ramos et al. [23], the proposed modified block Falkner method are consistent provided that the order of the methods is $p\geq 1$. The orders of our modified Falkner block methods are $p=4,5,6,$ hence the proposed methods are consistent. The sufficient conditions for the linear multistep method to be convergent are to be zero-stable and consistent. Since the two conditions are satisfied, we conclude that the new methods converge.

3.3. Linear stability analysis of the proposed methods

Following the guidelines in [25] we discuss the linear stability of the proposed methods considering the standard test equation introduced by Lambert and Watson [26]

$$y''=-\mu^2 y \tag{11}$$

where μ is a complex parameter. For simplicity we assume that the problem in (1) is a scalar one (in other case, for a system we would have to consider the Kronecker product of matrices to formulate the methods). On the uniform grid $x_n=x_0+ih, i=0,1,\ldots,N,$ the proposed methods may be reformulated in the form

$$\begin{aligned} Y^{[n]} &= A\,F\left(Y^{[n]}\right)+U\,y^{[n-1]}\\ y^{[n]} &= B\,F\left(Y^{[n]}\right)+V\,y^{[n-1]}, \end{aligned} \tag{12}$$

where

$$A\in R^{(k+2)\times(k+2)},\quad B\in R^{k\times(k+2)},\quad U\in R^{(k+2)\times k},\quad V\in R^{k\times k}$$

are matrices of coefficients according to a specific method, and

$$Y^{[n]}=\begin{pmatrix} y_n\\ y_{n+1}\\ \vdots\\ y_{n+k}\\ hy'_{n+k}\end{pmatrix},\qquad y^{[n]}=\begin{pmatrix} y_{n+2}\\ hy'_n\\ hy'_{n+2}\\ \vdots\\ hy'_{n+k-1}\end{pmatrix},\qquad F\left(Y^{[n]}\right)=\begin{pmatrix} h^2 f_n\\ h^2 f_{n+1}\\ \vdots\\ h^2 f_{n+k}\\ h^3 g_{n+k}\end{pmatrix}.$$

Applying (12) to (11), we obtain

$$Y^{[n]}=-\mu^2 h^2\,A\,Y^{[n]}+U\,y^{[n-1]} \tag{13}$$

$$y^{[n]}=-\mu^2 h^2\,B\,Y^{[n]}+V\,y^{[n-1]}. \tag{14}$$

Put $z=\mu^2 h^2$ and assume that the matrix $I+zA$ is nonsingular, where I is the identity matrix of same dimension of A. Then it follows from (13) that

$$Y^{[n]}=(I+zA)^{-1}\,U\,y^{[n-1]},$$

and substituting this relation into (14) we obtain that

$$y^{[n]}=M(z)\,y^{[n-1]}$$

where $M(z)$ is the so called stability matrix of the proposed methods, and is given by

$$M(z)=-zB(I+zA)^{-1}U+V.$$

The behavior of the numerical solution will depend on the eigenvalues of this matrix, and the stability properties of the methods will be characterized by the spectral radius, $\rho(M(z))$. According to the terminology introduced by Coleman and Ixaru [27] we have that $(0,\,z_s)$ is an interval of stability for a given method if for all $z\in(0,\,z_s)$ it is $|r_i|<1$ where the $r_i, i=1,\ldots,k$ are the eigenvalues of the stability matrix.

The region of absolute stability $\mathbb{A}$ is the set of all complex values $z\in\mathbb{C}$ such that all the eigenvalues of the stability matrix are inside the unit circle, that is,

$$\mathbb{A}=\{z\in\mathbb{C}:|r_i|<1, i=1,2,\ldots,k\}.$$

The intersection of $\mathbb{A}$ with the real axis gives the stability intervals.

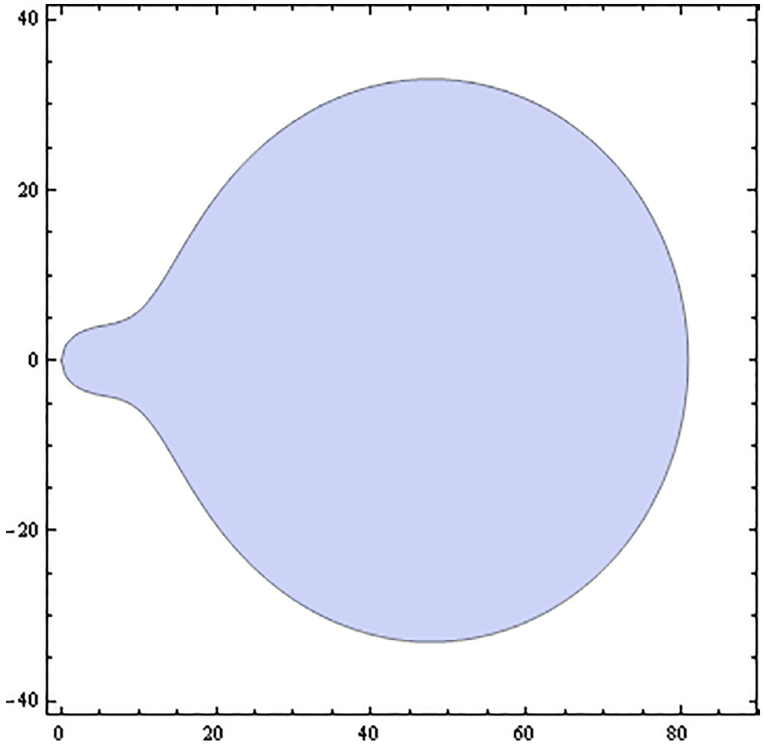

Fig. 1. Stability region for the 2-step Falkner block method.

Table 4
Stability intervals of the modified block Falkner methods.

k	Interval of stability
2	$\left(0, 12\left(3+\sqrt{14}\right)\right) \simeq (0, 80.8998)$
3	$\left(0, 2\sqrt{30}\right) \simeq (0, 10.9545)$
4	$\left(0, \frac{1}{4}\left(\sqrt{1921}-1\right)\right) \simeq (0, 10.7073)$

As an example, for $k = 3$ the modified Falkner method may be put in the form in (12) with

$$A = \begin{pmatrix} \frac{97}{1080} & \frac{131}{240} & -\frac{29}{120} & \frac{229}{2160} & -\frac{17}{360} \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{13}{1080} & \frac{5}{16} & \frac{11}{40} & -\frac{163}{2160} & \frac{11}{360} \\ -\frac{4}{135} & \frac{11}{15} & \frac{4}{3} & -\frac{1}{27} & \frac{2}{45} \\ -\frac{2}{135} & \frac{2}{5} & \frac{6}{5} & \frac{56}{135} & -\frac{2}{45} \end{pmatrix} \qquad U = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} -\frac{13}{1080} & \frac{5}{16} & \frac{11}{40} & -\frac{163}{2160} & \frac{11}{360} \\ -\frac{367}{1080} & -\frac{19}{20} & \frac{21}{40} & -\frac{127}{540} & \frac{19}{180} \\ -\frac{23}{1080} & \frac{9}{20} & \frac{29}{40} & -\frac{83}{540} & \frac{11}{180} \end{pmatrix} \qquad V = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

The stability regions of the methods for $k = 2, 3, 4$ are presented in Figs. 1–3, respectively. In Table 4 we have included the absolute stability intervals for the proposed methods. We have considered only these methods because for $k \geq 5$ we have found that the methods do not have primary stability intervals, that is, stability intervals of the form $(0, z_s)$, although other secondary stability intervals may exist.

4. Formulas of the methods for $k = 2, 3, 4$

In this section, we present the formulas corresponding to the modified Falkner methods for a number of steps $k = 2, 3, 4$. We note that there are many ways to give these formulas, depending on what terms we choose to place on the left side of the equations.

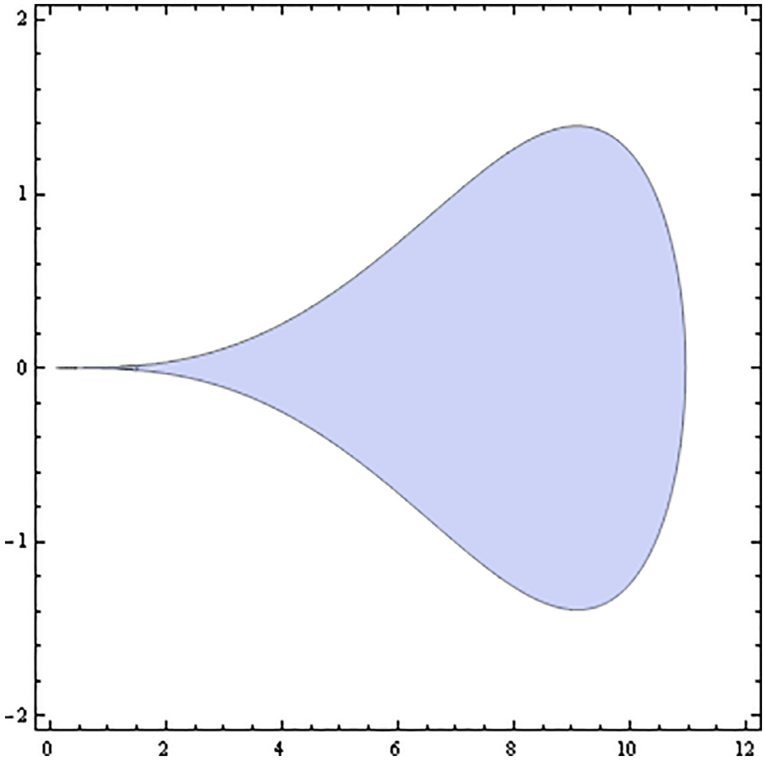

Fig. 2. Stability region for the 3-step Falkner block method.

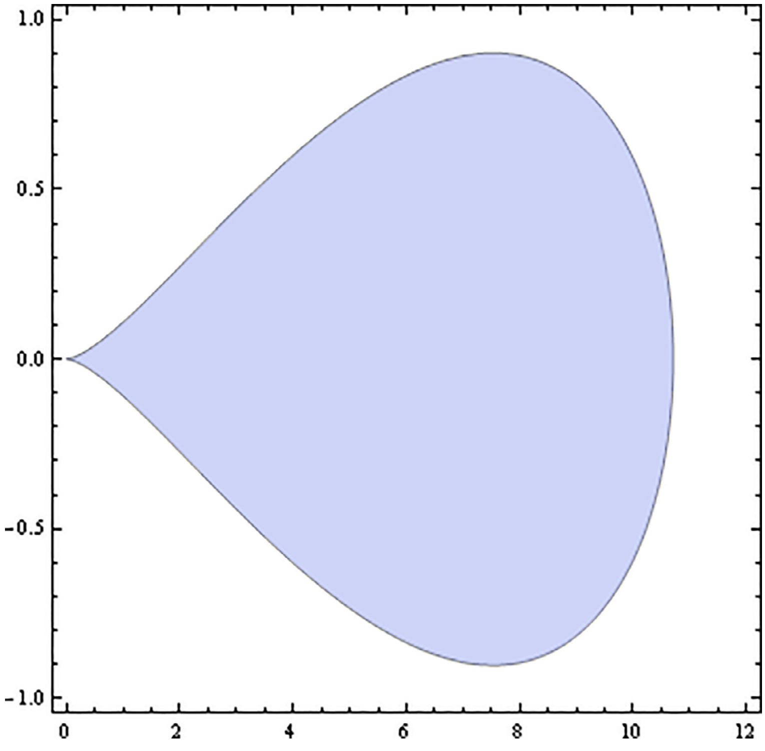

Fig. 3. Stability region for the 4-step Falkner block method.

4.1. $k = 2$ steps

$$
\begin{aligned}
y_{n+2} &= -\frac{7}{120}h^3 g_{n+2} + \frac{17}{80}h^2 f_{n+2} + \frac{3}{10}h^2 f_{n+1} - \frac{1}{80}h^2 f_n + hy'_{n+1} + y_{n+1},\\
y_n &= \frac{7}{120}h^3 g_{n+2} - \frac{31}{240}h^2 f_{n+2} + \frac{8}{15}h^2 f_{n+1} + \frac{23}{240}h^2 f_n - hy'_{n+1} + y_{n+1},\\
hy'_n &= -\frac{1}{8}h^3 g_{n+2} + \frac{13}{48}h^2 f_{n+2} - \frac{11}{12}h^2 f_{n+1} - \frac{17}{48}h^2 f_n + hy'_{n+1},\\
hy'_{n+2} &= -\frac{1}{8}h^3 g_{n+2} + \frac{29}{48}h^2 f_{n+2} + \frac{5}{12}h^2 f_{n+1} - \frac{1}{48}h^2 f_n + hy'_{n+1}
\end{aligned}
$$

4.2. $k = 3$ steps

$$
\begin{aligned}
y_{n+3} &= -\frac{4}{135}h^2 f_n + \frac{11}{15}h^2 f_{n+1} + \frac{4}{3}h^2 f_{n+2} - \frac{1}{27}h^2 f_{n+3} + \frac{2}{45}h^3 g_{n+3} + 2\,hy'_{n+1} + y_{n+1},\\
y_n &= -\frac{17}{360}h^3 g_{n+3} + \frac{229}{2160}h^2 f_{n+3} - \frac{29}{120}h^2 f_{n+2} + \frac{131}{240}h^2 f_{n+1} + \frac{97}{1080}h^2 f_n - hy'_{n+1} + y_{n+1},\\
y_{n+2} &= -\frac{13}{1080}h^2 f_n + \frac{5}{16}h^2 f_{n+1} + \frac{11}{40}h^2 f_{n+2} - \frac{163}{2160}h^2 f_{n+3} + \frac{11}{360}h^3 g_{n+3} + hy'_{n+1} + y_{n+1},\\
y'_n &= \frac{19}{180}h^2 g_{n+3} - \frac{127}{540}hf_{n+3} + \frac{21}{40}hf_{n+2} - \frac{19}{20}hf_{n+1} - \frac{367}{1080}hf_n + y_{n+1},\\
y'_{n+2} &= -\frac{23}{1080}hf_n + \frac{9}{20}hf_{n+1} + \frac{29}{40}hf_{n+2} - \frac{83}{540}hf_{n+3} + \frac{11}{180}h^2 g_{n+3} + y_{n+1},\\
y'_{n+3} &= -\frac{2}{135}hf_n + \frac{2}{5}hf_{n+1} + \frac{6}{5}hf_{n+2} + \frac{56}{135}hf_{n+3} - \frac{2}{45}h^2 g_{n+3} + y_{n+1}.
\end{aligned}
$$

4.3. $k = 4$ steps

$$
\begin{aligned}
y_n &= \frac{41}{1008}h^3 g_{n+4} - \frac{5839}{60480}h^2 f_{n+4} + \frac{289}{1260}h^2 f_{n+3} - \frac{41}{140}h^2 f_{n+2} + \frac{2179}{3780}h^2 f_{n+1} +\\
&\quad \frac{337}{4032}h^2 f_n - hy'_{n+1} + y_{n+1},\\
y_{n+2} &= -\frac{97}{10080}h^2 f_n + \frac{1139}{3780}h^2 f_{n+1} + \frac{97}{336}h^2 f_{n+2} - \frac{163}{1260}h^2 f_{n+3} + \frac{1481}{30240}h^2 f_{n+4} -\\
&\quad \frac{5}{252}h^3 g_{n+4} + hy'_{n+1} + y_{n+1},\\
y_{n+3} &= -\frac{59}{2520}h^2 f_n + \frac{664}{945}h^2 f_{n+1} + \frac{97}{70}h^2 f_{n+2} - \frac{10}{63}h^2 f_{n+3} + \frac{709}{7560}h^2 f_{n+4} - \frac{5}{126}h^3 g_{n+4}\\
&\quad + 2\,h' y_{n+1} + y_{n+1},\\
y_{n+4} &= -\frac{81}{2240}h^2 f_n + \frac{153}{140}h^2 f_{n+1} + \frac{729}{280}h^2 f_{n+2} + \frac{81}{140}h^2 f_{n+3} + \frac{117}{448}h^2 f_{n+4} - \frac{9}{112}h^3 g_{n+4} +\\
&\quad 3\,hy'_{n+1} + y_{n+1},\\
y'_n &= -\frac{3}{32}h^2 g_{n+4} + \frac{1277}{5760}hf_{n+4} - \frac{47}{90}hf_{n+3} + \frac{311}{480}hf_{n+2} - \frac{46}{45}hf_{n+1} - \frac{1873}{5760}hf_n + y_{n+1},\\
y'_{n+2} &= -\frac{97}{5760}hf_n + \frac{58}{135}hf_{n+1} + \frac{359}{480}hf_{n+2} - \frac{23}{90}hf_{n+3} + \frac{1639}{17280}hf_{n+4} - \frac{11}{288}h^2 g_{n+4} + y_{n+1},\\
y'_{n+3} &= -\frac{1}{90}hf_n + \frac{17}{45}hf_{n+1} + \frac{19}{15}hf_{n+2} + \frac{17}{45}hf_{n+3} - \frac{1}{90}hf_{n+4} + y_{n+1},\\
y'_{n+4} &= -\frac{9}{640}hf_n + \frac{2}{5}hf_{n+1} + \frac{189}{160}hf_{n+2} + \frac{9}{10}hf_{n+3} + \frac{341}{640}hf_{n+4} - \frac{3}{32}h^2 g_{n+4} + y_{n+1}
\end{aligned}
$$

5. Implementation details

The proposed method is implemented in a block mode, without the need of predictors. On each block interval of the form $[x_n, x_{N-k}]$, $n = 0, \ldots, N-k$, being N divisible by k in order to have an integer number of blocks, we solve the system in (5) (or in (6) in case the problem is of the special form $y'' = f(x, y)$) using the Newton's method. The starting values for applying Newton's method are taken as the approximations provided by the Taylor formulas

$$\begin{aligned} y_{n+i} &= y_n + hy'_n + \frac{h^2}{2} f_n \\ y'_{n+i} &= y'_n + hf_n, \quad i = 1, \dots, k\,. \end{aligned}$$

The presence of g_{n+k} in the formulas of the block method, which approximates the third derivative at x_{n+k}, that is, $g_{n+k} \simeq y'''(x_{n+k})$), requires the calculation of

$$y'''(x) = \frac{df(x, y, y')}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} y' + \frac{\partial f}{\partial y'} f,$$

which can be easily obtained by hand, or in more difficult cases, with the use of a computer algebra system.

Note that the proposed method may also be used for solving systems of second-order differential equations, by considering a component-wise implementation. For a system of m equations, given in vector form as

$$\mathbf{y}'' = \mathbf{f}(x, \mathbf{y}^T, \mathbf{y}'^T), \quad \mathbf{y}(a) = \mathbf{y}_0, \quad \mathbf{y}'(a) = \dot{\mathbf{y}}_0, \qquad a = x_0 \le x \le b = x_N$$

where $\mathbf{y} = (y_1, \dots, y_m)^T, \mathbf{y}' = (y'_1, \dots, y'_m)^T$,

$$\mathbf{f}(x, \mathbf{y}^T, \mathbf{y}'^T) = (f_1(x, \mathbf{y}^T, \mathbf{y}'^T), \dots, f_m(x, \mathbf{y}^T, \mathbf{y}'^T))^T,$$

and $\mathbf{y}_0 = (y_{1,0}, \dots, y_{m,0})^T$, $\mathbf{y}'_0 = (\dot{y}_{1,0}, \dots, \dot{y}_{m,0})^T$, we apply the method to each of the scalar equations in the differential system. In the general case this would result in an algebraic system of $2k \times m$ equations, that may be solved using again the Newton's method, as in the scalar case. To get the approximate values of the third derivative of each component at x_{n+k}, denoted by $g_{i,n+k} \simeq y'''_i(x_{n+k})$, $i = 1, \dots, m$, we use the formula

$$y'''_i(x) = \frac{df_i(x, y_1, \dots, y_m, y'_1, \dots, y'_m)}{dx} = \frac{\partial f_i}{\partial x} + \sum_{j=1}^{m} \frac{\partial f_i}{\partial y_j} y'_j + \sum_{j=1}^{m} \frac{\partial f_i}{\partial y'_j} f_j\,.$$

6. Numerical examples

In this section, the performance of the proposed methods is tested on some examples of second order IVPs in ODEs. In order to make fair comparisons with the results provided by some existing methods appeared in the literature, we have considered two types of errors:

- AE: absolute error at the point considered

$$AE(x_i) = |y(x_i) - y_i|\,.$$

- MAE: maximum absolute error on the grid points at the integration interval

$$MAE = \max_{x_j \in \{x_1, \dots, x_N\}} |y(x_j) - y_j|\,.$$

The proposed two-step, three-step, and four-step modified block Falkner methods will be named respectively as *MBF*2, *MBF*3 and *MBF*4. Discussion of the results of the new methods is also done here. From the numerical results, we can see that the proposed methods perform very well compared to some existing methods in the literature. The following problems are taken as test problems:

6.1. Problem 1

In the first problem, we consider the non-homogeneous IVP of ODE studied earlier by Jator and Li [16] and Olabode amd Momoh [28] which is given by

$$y''(x) - 4y'(x) + 8y(x) = x^3, \; y(0) = 2, \; y'(0) = 4, \; 0 \le x \le 1\,.$$

The exact solution is

$$y(x) = e^{2x}\left(2\cos(2x) - \frac{3}{64}\sin(2x)\right) + \frac{3x}{32} + \frac{3x^2}{16} + \frac{x^3}{8}\,.$$

Table 5 shows for $h = 0.1$ the absolute errors at different points on the integration interval using different methods, evincing the good performance of the proposed methods. The other methods used were the classical Runge–Kutta method of the fourth order, the Runge–Kutta method of Lobatto IIIA which has order six (see [29]), the continuous hybrid multistep method of fifth order in [28], and the self-starting block method of fifth order in [16].

In order to make a fair comparison, it is not enough to consider the errors but also the computational times used. Thus, we have included in Fig. 4 an efficiency plot showing the maximum absolute errors in logarithmic scale versus CPU times for the two methods of sixth order considered: the proposed *MBF*4 method and the Runge–Kutta Lobatto IIIA method. We see clearly that the proposed method performs better.

Table 5
Comparison of the absolute errors (AE) on Problem 1 taking $h = 0.1$.

x-value	AE with *MBF*2	*RK*4	AE with *MBF*3
0.1	1.61204×10^{-11}	2.12147×10^{-5}	1.18172×10^{-12}
0.2	1.99363×10^{-10}	3.97457×10^{-5}	1.00231×10^{-12}
0.3	8.20983×10^{-10}	4.77407×10^{-5}	9.31744×10^{-12}
0.4	2.30779×10^{-9}	3.37780×10^{-5}	2.75051×10^{-11}
0.5	5.28323×10^{-9}	1.76064×10^{-5}	6.02527×10^{-11}
0.6	1.06046×10^{-8}	1.26093×10^{-4}	1.13075×10^{-10}
0.7	1.93897×10^{-8}	3.15248×10^{-4}	1.92026×10^{-10}
0.8	3.30213×10^{-8}	6.11441×10^{-4}	3.03129×10^{-10}
0.9	5.31171×10^{-8}	1.04170×10^{-3}	4.51466×10^{-10}
1.0	8.14449×10^{-8}	1.63015×10^{-3}	6.3985×10^{-10}
x-value	*LobattoIIIA*	AE in [28]	AE in [16]
0.1	2.97638×10^{-9}	7.14261×10^{-8}	5.10704×10^{-6}
0.2	8.02910×10^{-9}	1.74900×10^{-7}	1.49586×10^{-5}
0.3	1.55131×10^{-8}	3.64488×10^{-7}	2.78532×10^{-5}
0.4	2.55642×10^{-8}	6.18978×10^{-7}	4.28908×10^{-5}
0.5	3.79330×10^{-8}	9.98898×10^{-7}	6.70307×10^{-5}
0.6	5.17730×10^{-8}	1.47940×10^{-6}	1.02637×10^{-4}
0.7	6.53837×10^{-8}	2.10220×10^{-6}	1.44907×10^{-4}
0.8	7.59166×10^{-8}	2.84088×10^{-6}	1.90905×10^{-4}
0.9	7.90619×10^{-8}	3.66899×10^{-6}	2.39733×10^{-4}
1.0	6.87467×10^{-8}	4.56167×10^{-6}	2.94670×10^{-4}

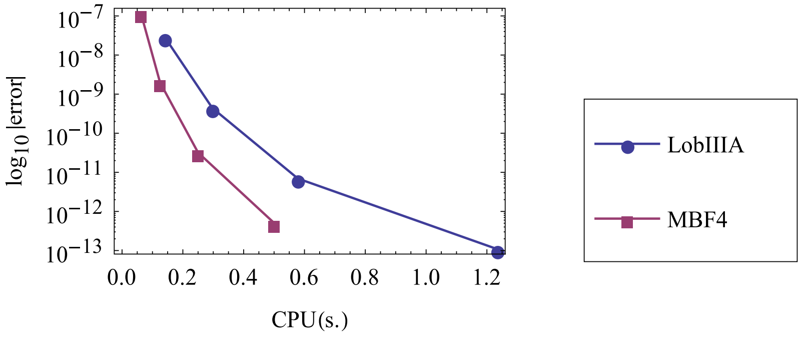

Fig. 4. Efficiency plot showing the maximum absolute errors (MAE) versus CPU times for Example 1.

6.2. Problem 2

In the second example we consider a dynamic IVP which was also solved by Areo and Omojola [30]. The problem is given as follows:

A 10 kg mass is attached to a spring having a spring constant of 140 N/m. The mass is started in motion from the equilibrium position with an initial velocity of 1 m/s in the upward direction and with an applied external force $F(t) = 5\sin(t)$. Find the subsequent motion of the mass if the force due to air resistance is $-a\dot{x}N$.

It follows from Newton's second law that

$$m\ddot{x} = -kx - a\dot{x} + F(t) \tag{15}$$

If the system begins at $t = 0$ with an initial velocity v_0 and from an initial position x_0, we also have the initial conditions

$$x(0) = x_0, \quad \dot{x}(0) = v_0\,. \tag{16}$$

Now if $m = 10, \quad k = 140, \quad a = 90$ and $F(t) = 5\sin(t)$ the equation of motion (15) becomes

$$\ddot{x} + 9\dot{x} + 14x = \frac{1}{2}\sin(t), \;\; 0 \leq t \leq 1\,. \tag{17}$$

According to Areo and Omojola [30], the exact solution to Eq. (17) with $x_0 = 0$ and $v_0 = -1$ is given by

$$x(t) = -\frac{9}{50}\,\mathrm{e}^{-2t} + \frac{99}{500}\,\mathrm{e}^{-7t} - \frac{9}{500}\cos(t) + \frac{13}{500}\sin(t)\,. \tag{18}$$

Note that the exponential terms, which come from the homogeneous solution, represent an associated free over-damped motion, which quickly dies out. These terms are the transient part of the solution. The terms coming from the particular solution, however, do not die out at $t \to \infty$; they are the steady-state part of the solution.

Table 6
Comparison of the absolute errors (AE) on Problem 2 taking $h = 0.1$.

t-value	AE with *MBF*2	AE with *MBF*3	AE in [30]
0.1000	5.89317×10^{-10}	2.61477×10^{-11}	1.274442×10^{-8}
0.2000	4.33459×10^{-9}	5.11038×10^{-11}	3.044226×10^{-8}
0.3000	7.26849×10^{-9}	1.39084×10^{-10}	4.150135×10^{-8}
0.4000	8.69447×10^{-9}	2.18257×10^{-10}	4.538448×10^{-8}
0.5000	8.92535×10^{-9}	2.92214×10^{-10}	4.429806×10^{-8}
0.6000	8.42496×10^{-9}	3.70645×10^{-10}	4.046609×10^{-8}
0.7000	7.56164×10^{-9}	4.64096×10^{-10}	3.547450×10^{-8}
0.8000	6.57436×10^{-9}	5.83087×10^{-10}	3.028463×10^{-8}
0.9000	5.59942×10^{-9}	7.38712×10^{-10}	2.540758×10^{-8}
1.0000	4.70495×10^{-9}	9.43678×10^{-10}	2.107144×10^{-8}
FCN	200	170	600

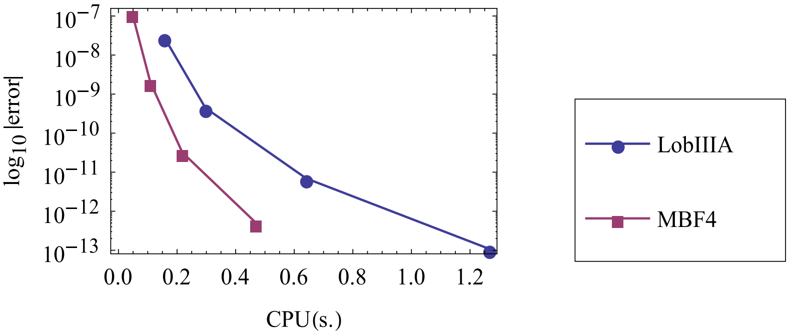

Fig. 5. Efficiency plot showing the maximum absolute errors (MAE) versus CPU times for Example 2.

Table 7
Comparison of the absolute errors (AE) at $x_N = 300$ on Problem 3.

h	Method in [31]	*MBF*4	LobattoIIIA
1	8.3×10^{-3} [1200]	8.4×10^{-2} [450]	1.0×10^{-4} [1800]
0.5	5.4×10^{-4} [2400]	1.0×10^{-3} [900]	1.6×10^{-6} [3600]
0.25	3.1×10^{-5} [4800]	1.2×10^{-5} [1800]	2.7×10^{-8} [7200]
0.125	1.8×10^{-6} [9600]	1.2×10^{-7} [3600]	2.5×10^{-9} [14400]
0.0625	1.1×10^{-7} [19200]	3.3×10^{-9} [7200]	2.1×10^{-9} [28800]

Table 6 shows when $h = 0.01$ the absolute errors for different methods at different points of the integration interval. The method used for comparisons was the fifth-order hybrid block method in [30]. The last row corresponds to the number of function evaluations used by each method to reach the final point of the integration interval.

Fig 5 shows an efficiency plot of the maximum absolute errors in logarithmic scale versus CPU times for the sixth order methods: *MBF*4 and the Runge–Kutta Lobatto IIIA. We see again that the proposed method clearly performs better.

6.3. Problem 3

The following problem is a nonlinear Duffing equation which was also solved by Simos in [31], and is given by

$$y'' + y + y^3 = B\cos(\Psi x), \quad y(0) = C_0, \quad y'(0) = 0\,.$$

The analytical solution of this problem is

$$y(x) = C_1\cos(\Psi x) + C_2\cos(3\Psi x) + C_3\cos(5\Psi x) + C_4\cos(7\Psi x),$$

where $\Psi = 1.01$, $B = 0.002$, $C_0 = 0.200426728069$, $C_1 = 0.200426728069$, $C_2 = 0.246946143 \times 10^{-3}$, $C_3 = 0.304016 \times 10^{-6}, C_4 = 0.374 \times 10^{-9}$. The integration interval considered was [0,300], as in [31]. In Table 7 we present the absolute error at the final point for different step-sizes. The numbers between brackets refer to the number of function evaluations used for each method. Fig. 6 shows the efficiency plot of the absolute global errors in logarithmic scale versus CPU times for the sixth order methods: *MBF*4 and the Runge–Kutta Lobatto IIIA.

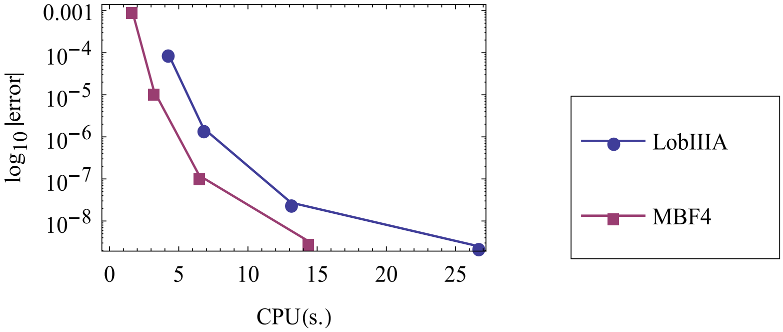

Fig. 6. Efficiency plot showing the maximum absolute errors (MAE) versus CPU times for Example 3.

Table 8
Comparison of the absolute errors (AE) on Problem 4.

x-value	Method in [32]	LobattoIIIA	*MBF*4
	h=1/30		h=1/60
1.1	9.44×10^{-10}	1.23×10^{-13}	5.32×10^{-14}
1.3	8.51×10^{-9}	4.00×10^{-13}	3.17×10^{-13}
1.5	2.41×10^{-8}	7.49×10^{-13}	7.95×10^{-13}
1.7	4.86×10^{-8}	1.19×10^{-12}	1.49×10^{-12}
1.9	8.30×10^{-8}	1.75×10^{-12}	2.44×10^{-12}
2.0	1.04×10^{-7}	2.08×10^{-12}	3.00×10^{-12}

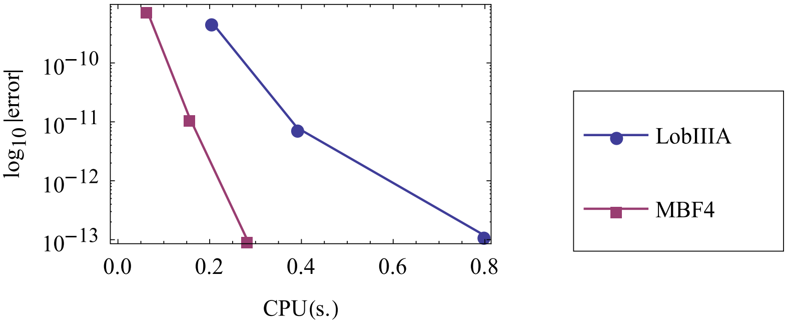

Fig. 7. Efficiency plot showing the maximum absolute errors (MAE) versus CPU times for Example 4.

6.4. Problem 4

In the fourth problem we consider a Cauchy–Euler non-homogeneous linear ODE which was also solved by Jator [32] given by

$$x^2 y'' - 3xy' + 3y = 2x^3 - x^2, \quad y(1) = 2, \quad y'(1) = 10,$$

which has been integrated in [1,2]. The analytical solution of this problem is

$$y(x) = 3x^3 - 2x + x^2(1 + x \ln x).$$

In Table 8, we present the absolute errors at different points taking $h = 1/30$ for the method with $k = 3$ in [32] and for the Lobatto IIIA method, while for the *MBF*4 we took $h = 1/60$. These different values of the step-sizes have been taken in order to have a fair comparison between the *MBF*4 and the Lobatto IIIA methods. In this way, being *MBF*4 a two-step block method, and the Lobatto IIIA a one-step method, the number of systems to be solved is the same. We see in Table 8 that the errors provided by the two methods are similar, but the *MBF*4 used 0.156 s of CPU time, while the Lobatto IIIA used 0.546 s. This better performance can be seen in Fig. 7 where we have plotted the maximum absolute errors in logarithmic scale versus CPU times for the two methods of sixth order considered.

6.5. Problem 5

Consider the following linear system of second order ordinary differential equations, which was also solved by Ramos et al. [33],

Table 9
Comparison of the maximum absolute errors (MAE) on Problem 5.

N	Method	MAE $y_1(x)$	MAE $y_1'(x)$	MAE $y_2(x)$	MAE $y_2'(x)$
102	*MBF*3	3.03199×10^{-12}	4.72805×10^{-11}	7.32653×10^{-11}	1.47581×10^{-10}
103	RFalkner3(2)	9.70370×10^{-10}	2.09210×10^{-9}	1.01360×10^{-8}	3.10460×10^{-8}
254	ode23s	8.6903×10^{-7}	1.73800×10^{-6}	1.30930×10^{-6}	3.92790×10^{-6}
1175	ode23s	4.05330×10^{-8}	8.10670×10^{-8}	6.09190×10^{-8}	1.82570×10^{-7}

Table 10
Comparison of the maximum absolute errors (MAE) on Problem 6.

Method	N	FCN	MAE
2PASDIR	490	2900	1.68517×10^{-9}
BFM$_6$	490	1134	7.13663×10^{-10}
*MBF*2	96	192	3.00131×10^{-11}
*MBF*3	96	160	1.99300×10^{-13}
*MBF*4	96	144	1.36933×10^{-15}

$$\begin{aligned} &y_1''(x) = 4y_1(x),\ y_1(0) = 1,\ y_1'(0) = -2, \\ &y_2''(x) = 9y_2(x),\ y_2(0) = 1,\ y_2'(0) = -3, \qquad 0 \le x \le 1 \end{aligned}$$

The exact solution of the problem is given by:

$$y_1(x) = \mathrm{e}^{-2x}, \qquad y_2(x) = \mathrm{e}^{-3x}.$$

In Table 9, we can see the good performance of the proposed methods, where N refers to the number of integration intervals. The other methods used for comparison were the third-order variable step-size rational Falkner method RFalkner3(2) in [33] and the well-known *ode*23*s* solver of the Matlab environment [34].

6.6. Problem 6

Finally, we consider the following initial-value system of second order ordinary differential equations. This problem was also solved by Ramos et al. [23] and Majid et al. [35], and is given by

$$\begin{aligned} &y_1'' = \frac{-y_1}{\sqrt{y_1^2 + y_2^2}},\ y_1(0) = 1,\ y_1'(0) = 0, \\ &y_2'' = \frac{-y_2}{\sqrt{y_1^2 + y_2^2}},\ y_2(0) = 0,\ y_2'(0) = 1, \qquad x \in [0, 1]. \end{aligned}$$

The theoretical solution of the problem is given by

$$y_1(x) = \cos(x), \qquad y_2(x) = \sin(x).$$

The maximum absolute errors with the proposed methods and the methods 2PASDIR (seventh order block method implemented in a variable step-size mode in [35]), and *BFM*$_6$ (seventh order block Falkner method in [23]) are presented in Table 10. We can see again the proposed methods outperform the methods used for comparison, even being the last ones of a higher order.

6.7. Discussion of results

This section deals with the explanation of the results. The data of the proposed modified block Falkner methods are presented in Tables $(4 - 9)$. The new methods have been used to solve different problems and the numerical results are compared with the results of existing methods, like the classical Runge–Kutta method of fourth order, the Runge–Kutta method of Lobatto IIIA which has order six (see [29]), the block method proposed by Jator and Li [16], the multistep methods reported by Vigo-Aguiar and Ramos [6], the rational Falkner-type method developed by Ramos et al. [23], the hybrid block approach proposed by Areo and Omojola [30], the direct implicit block method proposed by Majid et al. [35]. The numerical results shown in Tables $(4 - 9)$ reveal that the new proposed modified Falkner methods perform better than the existing methods used for comparisons.

7. Conclusions

In this manuscript, we have developed a family of modified k-step block Falkner methods using the third derivative, for solving general second-order initial value problem in ODEs directly without the need to convert it to an equivalent system of first order ODEs. The characteristics of the methods have been studied. We have used the new methods to solve different problems which have appeared in the literature. The comparison of the errors shows that the new methods perform very much better than some existing methods. The modified block Falkner methods proposed in this paper have been tested for its usability and therefore they may be considered for solving second-order initial value problems in ODEs.

Acknowledements

The authors are grateful to two referees for their careful reading of the manuscript and for pointing out some comments which helped to improve the final result.

References

[1] J.D. Lambert, Computational Methods in Ordinary Differential Equations, John Wiley and Sons, London, 1973.

[2] J.D. Lambert, Numerical Methods for Ordinary Differential Systems, John Wiley and Sons, New York, 1991.

[3] P. Onumanyi, D.O. Awoyemi, S.N. Jator, U.W. Sirisena, New linear multistep methods with continuous coefficients for first order initial value problems, J. Niger. Math. Soc. 13 (1994) 37–51.

[4] R. Jeltsch, L. Kratz, On the stability properties of Brown's multistep multi–derivative methods, Numerische Mathematik 30 (1978) 25–38.

[5] D.O. Awoyemi, A class of continuous methods for general second order IVPs in ordinary differential equations, Int. J. Comput. Math. 72 (1999) 29–37.

[6] J. Vigo-Aguiar, H. Ramos, Variable stepsize implementation of multistep methods for $y'' = f(x, y, y')$, J. Comput. Appl. Math. 192 (2006) 114–131.

[7] F. Mazzia, A.M. Nagy, A new mesh selection strategy with stiffness detection for explicit Runge–Kutta methods, Appl. Math. Comput. 255 (2015) 125–134.

[8] F. Mazzia, J.R. Cash, K. Soetaert, A test set for stiff initial value problem solvers in the open source software r: Package DeTestSet, J. Comput. Appl. Math. 236 (2012) 4119–4131.

[9] B.P. Sommeijer, Explicit, high–order Runge–Kutta–Nyström methods for parallel computers, Appl. Numer. Math. (13) (1993) 221–240.

[10] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, 1998.

[11] J.C. Butcher, G. Hojjati, Second derivative methods with Runge–Kutta stability, Numer. Algorithms 40 (2005) 415–429.

[12] J. Franco, Runge–Kutta–Nyström methods adapted to the numerical integration of perturbed oscillators, Comput. Phys. Commun. 147 (2002) 770–787.

[13] S.M. Mahmoud, M.S. Osman, On a class of spline-collocation methods for solving second-order initial-value problems, Int. J. Comput. Math. 86 (2009) 613–630.

[14] S.N. Jator, Solving second order initial value problems by a hybrid multistep method without predictors, Appl. Math. Comput. 217 (2010) 4036–4046.

[15] A.O. Adeniran, B.S. Ogundare, An efficient hybrid numerical scheme for solving general second order initial value problems (IVPs), Int. J. Appl. Math. Res. 4 (2) (2015) 411–419.

[16] S.N. Jator, J. Li, A self-starting linear multistep method for a direct solution of the general second order initial value problem, Int. J. Comput. Math. 86 (2009) 827–836.

[17] H. Ramos, C. Lorenzo, Review of explicit Falkner methods and its modifications for solving special second-order IVPs, Comput. Phys. Commun. 181 (2010) 1833–1841.

[18] H. Ramos, M.F. Patricio, Some new implicit two-step multiderivative methods for solving special second-order IVP's, Appl. Math. Comput. 239 (2014) 227–241.

[19] L.K. Yap, F. Ismail, N. Senu, An accurate block hybrid collocation method for third order ordinary differential equations, J. Appl. Math. 2014 (2014) Article ID 549597, 9 pages, doi:10.1155/2014/549597.

[20] H. Ramos, Z. Kalogiratou, T. Monovasilis, T.E. Simos, An optimized two-step hybrid block method for solving general second order initial-value problems, Numer. Algorithms 72 (4) (2016) 1089–1102.

[21] M.A. Rufai, E.A. Areo, An efficient one-eight step hybrid block method for solving second order initial value problems of ODEs, Int. J. Differ. Equ. Appl. 15 (2) (2016) 117–139.

[22] J. Li, X. Wu, Adapted Falkner-type methods solving oscillatory second–order differential equations, Numer. Algorithms 62 (2013) 355–381.

[23] H. Ramos, S. Mehta, J. Vigo-Aguiar, A unified approach for the development of k-step block Falkner-type methods for solving general second-order initial-value problems in ODEs, J. Comput. Appl. Math. 318 (2017) 550–564.

[24] S.O. Fatunla, Block methods for second order odes, Int. J. Comput. Math. 41 (1991) 55–63.

[25] Z. Jackiewicz, General Linear Methods for Ordinary Differential Equations, John Wiley and Sons, New Jersey, 2009.

[26] J.D. Lambert, I.A. Watson, Symmetric multistep methods for periodic initial value problems, IMA J. Appl. Math. 18 (1976) 189–202.

[27] J.P. Coleman, L.G. Ixaru, P-stability and exponential-fitting methods for $y'' = f(x, y)$, IMA J. Numer. Anal. 16 (1996) 179–199.

[28] B.T. Olabode, A.L. Momoh, Continuous hybrid multistep methods with Legendre basis function for direct treatment of second order stiff ODEs, American J. of Computational and Applied Mathematics 6 (2) (2016) 38–49.

[29] J.C. Butcher, Numerical Methods for Ordinary Differential Equations, John Wiley and Sons, 2008.

[30] E.A. Areo, M.O. Omojola, A new one-twelfth step continuous block method for the solution of modeled problems of ordinary differential equations, Am. J. Comput. Math. 5 (2015) 447–457.

[31] T.E. Simos, An exponentially-fitted Runge–Kutta method for the numerical integration of initial-value problems with periodic or oscillating solutions, Comput. Phys. Commun. 115 (1998) 1–8.

[32] S.N. Jator, A class of initial value methods for the direct solution of second order initial value problems, Int. J. Pure Appl. Math. 46 (2) (2008) 225–230.

[33] H. Ramos, G. Singh, V. Kanwar, S. Bhatia, An efficient variable step-size rational Falkner-type method for solving the special second-order IVP, Appl. Math. Comput. 291 (2016) 39–51.

[34] R. Ashino, M. Nagase, R. Vaillancourt, Behind and beyond the Matlab ODE suite, Comput. Math. Appl. 40 (2000) 491–512.

[35] Z.A. Majid, N.A. Azmi, M. Suleiman, Solving second order ordinary differential equations using two point four step direct implicit block method, Eur. J. Sci. Res. 31 (2009) 29–36.